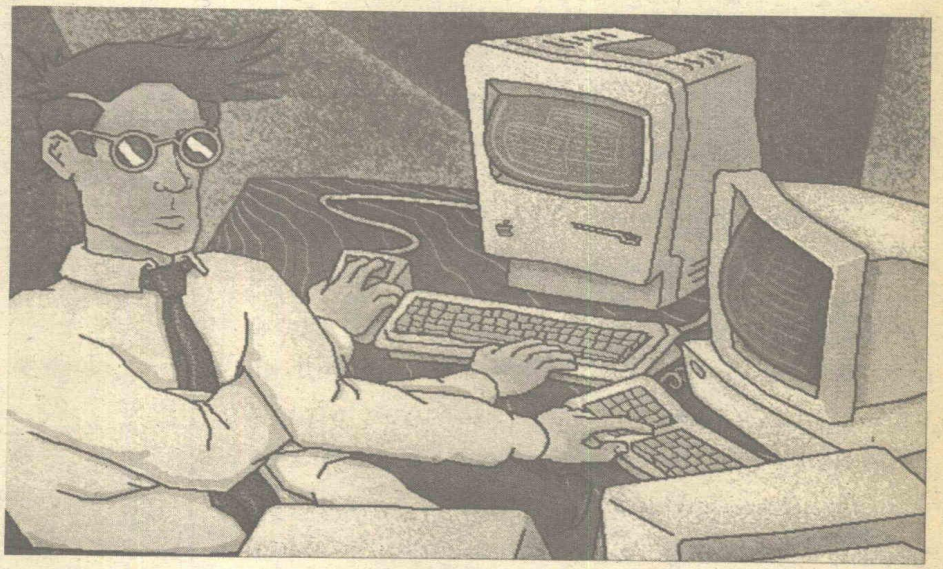
ชัยยงค์ วงศ์ชัยสุวัฒน์

การใช้โปรแกรม DOSKEY ของดอส 5.0



นำเสนอโปรแกรม DOSKEY ที่ติดมากับเอ็มเอสดอส 5.0 เก่งอย่างไรน่ะ

นับตั้งแต่ดอสเวอร์ชันแรก ๆ มาแล้ว เราสามารถเรียกคำสั่งเดิมล่าสุดที่เพิ่งใช้ผ่านไปกลับมาแก้ไขใหม่ด้วยการใช้ปุ่มกดแก้ไข ตัวอย่างเช่น เมื่อเราใช้คำสั่งคัดลอกไฟล์ COPY ทำการคัดลอกไฟล์ต้นทางไปเป็นไฟล์ปลายทาง แต่บังเอิญพิมพ์ชื่อไฟล์ต้นทางผิดทำให้ค้นหาไฟล์นั้นไม่พบ แทนที่เราจะพิมพ์คำสั่งเข้าไปใหม่ เราสามารถใช้ปุ่มกดแก้ไขที่ประกอบด้วยปุ่มฟังก์ชันคีย์ F1 ถึง F6 ปุ่ม DEL, INS, ESC ปุ่มลูกศรซ้ายขวา และปุ่มถอยกลับ BS เรียกคำสั่งเดิมมาแก้ไขใหม่แล้วส่งคำสั่งแก้ไขแล้วไปทำงานด้วยการกดปุ่ม Enter

ในเอ็มเอสดอสเวอร์ชัน 5.0 หรือในดอสฉบับ DR DOS 5.0 ได้เพิ่มเติมการใช้งานส่วนนี้ให้มีขีดความสามารถสูงขึ้นไปอีก ด้วยการเก็บคำสั่งต่าง ๆ ที่เรียกใช้ไปแล้วเป็นประวัติรายการ ซึ่งทำให้เราทราบว่าใช้คำสั่งอะไรไปบ้าง ขณะเดียวกันสามารถเรียกคำสั่งในอดีตกลับมาใช้งานโดยไม่ต้องพิมพ์คำสั่งเข้าไปใหม่ หรือเรียกคำสั่งเดิมในอดีตมาทำการแก้ไขแล้วส่งทำงานใหม่

มุมโอเอสฉบับนี้จะกล่าวถึงโปรแกรม DOSKEY ที่มากับเอ็มเอสดอส 5.0 โปรแกรมนี้จะสงวนพื้นที่หน่วยความจำสำหรับเก็บบันทึกรายการคำสั่งที่ใช้ไปแล้ว ขณะเดียวกันจะใช้เพื่อสร้างโปรแกรมมาโครที่คล้าย ๆ กับแบตช์ไฟล์ โดยโปรแกรมมาโครจะเก็บในหน่วยความจำ แต่แบตช์ไฟล์จะเก็บไว้เป็นไฟล์ในดิสก์ ดังนั้นโปรแกรมมาโครจะทำงานได้รวดเร็วกว่า

ในตอนแรกของบทความจะเป็นการเสนอวิธีการเรียกใช้ปุ่มกดแก้ไขคำสั่งเดิม แล้วจึงตามด้วยการรันโปรแกรม DOSKEY เพื่อใช้งานเรียกและนำประวัติรายการคำสั่งเดิมมาแก้ไข และในตอนท้ายจะเสนอการใช้โปรแกรม DOSKEY มาสร้างโปรแกรมมาโคร

ปุ่มกดแก้ไขคำสั่ง

เมื่อพิมพ์คำสั่งดอสหลังเครื่องหมายเตรียมพร้อมรับคำสั่ง เช่น คำสั่ง COPY TEST1.TXT, TEST2.TXT แล้วตามด้วยการกดปุ่ม Enter เอ็มเอสดอสจะตีความคำสั่งและทำงานของขั้นตอนของคำสั่ง ขณะเดียวกันจะมีการเก็บบันทึกข้อมูลตัวอักษรต่าง ๆ ของคำสั่งนั้นลงในบัฟเฟอร์ ซึ่งเป็นพื้นที่หน่วยความจำสงวนเฉพาะไว้ ในบางครั้งเราอาจเรียกว่า template

ข้อมูลในบัฟเฟอร์จะเก็บเฉพาะคำสั่งใช้งานล่าสุดเท่านั้น ถ้าเราพิมพ์ใช้งานคำสั่งใหม่ ข้อมูลคำสั่งใหม่จะไปแทนที่ข้อมูลคำสั่งเดิมทันที โดยสามารถเรียกคำสั่งที่เก็บในบัฟเฟอร์นั้นมาทำการแก้ไขเรียกใช้งานใหม่ โดยไม่ต้องเสียเวลาพิมพ์คำสั่งยาว ๆ ใหม่ทั้งหมด

ในการเรียกคำสั่งในบัฟเฟอร์มาทำการแก้ไข จะเรียกผ่านทางปุ่มกดแก้ไขคำสั่ง (Editing Keys) ซึ่งจะประกอบด้วยปุ่มกดต่าง ๆ หลายปุ่ม ดังแสดงไว้ในรูปที่ 1

ตัวอย่างการใช้ปุ่มกดแก้ไขคำสั่ง

เพื่อให้เห็นรูปแบบของการใช้ปุ่มกดแก้ไขคำสั่งนี้ จึงขอยกตัวอย่างการใช้งานบางส่วนไว้ดังนี้

1. หลังจากการใช้คำสั่ง COPY C: \ TEMP \ *.DPC a: เพื่อคัดลอกไฟล์จากไดรฟ์ C ไปไดรฟ์ A พบว่าไม่มีไฟล์นามสกุล .DPC อยู่เลย เนื่องจากการพิมพ์นามสกุลไฟล์ผิดที่ถูกต้องจะต้องเป็นไฟล์นามสกุล .DOC เราสามารถเรียกคำสั่งในบัฟเฟอร์มาแก้ไขด้วยวิธี กดปุ่มฟังก์ชันคีย์ F3 ซึ่งจะได้คำสั่งเหมือนเดิม จากนั้นจึงกดปุ่มถอยกลับ BS หรือปุ่มลูกศรทางซ้ายลบข้อมูลทางซ้ายของเคอร์เซอร์เรื่อยมาจนถึงตัวอักษร P เหลือตัวอักษรตั้งต้นจนถึงตัวอักษร D ต่อไปพิมพ์ตัวอักษร O แล้วกดปุ่มฟังก์ชันคีย์ F3 แสดงส่วนที่เหลือ

2. จากตัวอย่างในข้อ 1 เราอาจเรียกคำสั่งในบัฟเฟอร์มาทีละตัวอักษรด้วยการใช้ปุ่มฟังก์ชันคีย์ F1 หลาย ๆ ครั้ง จนบรรทัดคำสั่งปรากฏเป็น COPY C: \ TEMP \ *.D เราจะพิมพ์ตัวอักษร O ต่อทันที กรณีนี้ตัวอักษร O จะไปแทนที่ตัวอักษร P ที่เหลือต่อไปจะเหมือนในคำสั่งเดิม จึงใช้ปุ่มฟังก์ชันคีย์ F3 เรียกตัวอักษรที่เหลือทั้งหมดมาต่อท้าย

3. จากตัวอย่างในข้อ 1 อีกเช่นกัน อาจเรียกคำสั่งในบัฟเฟอร์มาแก้ไข นับจากตั้งต้นมาจนถึงตัวอักษรที่ต้องการทันทีด้วยการใช้ปุ่มฟังก์ชันคีย์ F2 ในที่นี้จะเป็นการกดปุ่มฟังก์ชันคีย์ F2 แล้วตามด้วยปุ่ม D ที่บรรทัดคำสั่งจะปรากฏใน COPY C: \ TEMP \ *. แล้วพิมพ์ตัวอักษร DO ตามด้วยการกดปุ่มฟังก์ชันคีย์ F3 ในตัวอย่างนี้เราจะกดปุ่มฟังก์ชันคีย์ F2 ตามด้วยตัวอักษร P เพื่อหวังให้แสดง COPY C: \ TEMP \ *.D เลยไม่ได้ เพราะที่บรรทัดจะแสดงเพียง CO ของคำสั่ง COPY เท่านั้น (เราอาจใช้ปุ่มกดหลายอย่างผสมกัน เช่น กดปุ่มฟังก์ชันคีย์ F2 ตามด้วยตัวอักษร D แล้วกดปุ่มฟังก์ชันคีย์ F1 เรียกตัวอักษร D มาแสดงพิมพ์ตัวอักษร O ตามด้วยการกดปุ่มฟังก์ชันคีย์ F3 เรียกที่เหลือมาต่อท้าย เป็นต้น)

ปุ่มกด	คำอธิบาย
F1 หรือปุ่มลูกศรทางขวา	การเรียกคัดลอกข้อมูลคำสั่งในบัฟเฟอร์มาปรากฏที่จอภาพครางละตัวอักษร การกดปุ่มฟังก์ชันคีย์ F1 หรือปุ่มลูกศรทางขวาแต่ละครั้ง จะแสดงตัวอักษรทีละตัว
F2 ตามด้วยตัวอักษร	การคัดลอกข้อมูลคำสั่งเดิมจากบัฟเฟอร์มาปรากฏที่จอภาพด้วยการแสดงข้อมูลบัฟเฟอร์ นับจากเริ่มต้นมาจนถึงตำแหน่งที่ตัวอักษรที่กำหนดไว้ หลังการกดปุ่มฟังก์ชันคีย์ F2 โดยไม่แสดงตัวอักษรที่กำหนด เช่น การกดปุ่มฟังก์ชันคีย์ F2 ตามด้วย I เราจะได้คำสั่งเดิมคืนมาเป็น COPY TEST ไม่รวมตัวอักษร I
F3	การคัดลอกข้อมูลคำสั่งเดิมที่เหลืออยู่มาปรากฏที่จอภาพ
F4 ตามด้วยตัวอักษร	การลบข้อมูลคำสั่งเดิมจากบัฟเฟอร์ โดยจะลบจากตำแหน่งเริ่มต้นไปจนถึงตำแหน่งที่ระบุด้วยตัวอักษร หลังจากกดปุ่มฟังก์ชันคีย์ F4 และจะไม่รวมตัวอักษรที่ระบุนั้น เช่น การกดปุ่มฟังก์ชันคีย์ F4 ตามด้วย 1 (ลบ COPY TEST ออกไป ที่เหลือเป็น 1.TXT TEST2.TXT) ถ้ากดปุ่มฟังก์ชันคีย์ F3 ตามมาเลยจะได้ 1.TXT TEST2.TXT ปรากฏที่หน้าจอ
F5	การคัดลอกคำสั่งที่อยู่ต่อท้ายการเตรียมพร้อมรับคำสั่งดอสไปสู่บัฟเฟอร์ โดยไม่มีการทำงานตามคำสั่งนั้น
F6	การเรียกใช้ตัวอักษร CTRL + Z (Z) เพื่อบรรจุในบรรทัดคำสั่ง
ปุ่มลูกศรทางซ้าย หรือปุ่มถอยกลับ BS	การลบตัวอักษรที่อยู่ทางซ้ายของเคอร์เซอร์ของบรรทัดคำสั่ง โดยไม่มีผลต่อคำสั่งที่อยู่ในบัฟเฟอร์
DEL	การลบตัวอักษรในบัฟเฟอร์หรือ template ตรงตำแหน่งที่สอดคล้องกับตำแหน่งเคอร์เซอร์ของบรรทัดคำสั่งนั้น
INS	การเรียกสู่โหมดแทรก เพื่อว่าตัวอักษรที่พิมพ์เข้าไปจะไปแทรกที่ตำแหน่งเคอร์เซอร์แทนการแทนที่ควรกดปุ่มนี้อีกครั้งเพื่อหยุดโหมดแทรกไปสู่โหมดแทนที่ที่มีอยู่เดิม
ESC	การยกเลิกคำสั่งที่ปรากฏในบรรทัดคำสั่ง โดยไม่มีการเปลี่ยนแปลงแก้ไขข้อมูลบันทึกในบัฟเฟอร์

รูปที่ 1 รายการปุ่มกดแก้ไขคำสั่งสมมติข้อมูลในบัฟเฟอร์เป็น COPY TEST1.TXT, TEST2.TXT

4. ตัวอย่างต่อไปเป็นการพิมพ์ไฟล์ออกหน้าจอจากไดเรคทอรีย่อยผิด เช่น TYPE C:\TEMP\TEST.DOC ที่ถูกต้องจะต้องมาจากไดเรคทอรีลึกลงไปอีกชั้นหนึ่งคือ TYPE C:\TEMP\REPORT\TEST.DOC กรณีนี้เราจะต้องแทรก \REPORT มาต่อท้ายไดเรคทอรี \TEMP เดิม ด้วยการใช้ปุ่ม INS การเรียกคำสั่งท่อนแรกมาแสดงจะใช้การกดปุ่มฟังก์ชันคีย์ F2 ตามด้วยตัวอักษร M ได้ TYPE C:\TE กดปุ่มฟังก์ชันคีย์ F1 อีก 2 ครั้ง จากนั้นกดปุ่ม INS เพื่อแทรกมาพิมพ์ \REPORT แล้วจึงตามด้วยการกดปุ่มฟังก์ชันคีย์ F3 (ในข้อนี้เราอาจเรียก TYPE C:\TEMP มาแสดงด้วยการกดปุ่มฟังก์ชันคีย์ F2 ตามด้วยจำนวนสองครั้งติดต่อกันก็ได้)

5. สมมติเราต้องการพิมพ์ไฟล์ดังตัวอย่างในข้อ 4 แต่เกิดพิมพ์เพียง TYPE C:\TEMP\TEST.DOC ที่เรารู้ตัวว่าไม่ถูกต้อง เราสามารถส่งข้อความคำสั่งนั้นไปเก็บในบัฟเฟอร์ก่อนด้วยการกดปุ่มฟังก์ชันคีย์ F5 ที่หน้าจอจะแสดงตัวอักษร @ ต่อท้ายคำสั่งของการยกเลิก และขึ้นบรรทัดใหม่พร้อมรับคำสั่งทันที เราสามารถแก้ไขข้อมูลในบัฟเฟอร์ ดังตัวอย่างข้อ 4 ได้ต่อไป

6. หากเราพิมพ์คำสั่งเข้าไปในบรรทัดคำสั่งแล้ว มารู้ว่าเป็นคำสั่งที่พิมพ์ในภายหลัง เราสามารถยกเลิกคำสั่งด้วยการกดปุ่ม ESC โดยไม่ทำให้ข้อมูลในบัฟเฟอร์เดิมเปลี่ยนแปลง การยกเลิกด้วยการกดปุ่ม Enter หรือเรียกไปทำงานจะเป็นการเก็บข้อมูลใหม่ทันที ถ้ามีการยกเลิกคำสั่งด้วยปุ่ม ESC แล้ว ที่หน้าจอจะแสดงตัวอักษร \ ต่อท้ายคำสั่งยกเลิก และขึ้นบรรทัดใหม่พร้อมรับคำสั่งใหม่ต่อไป

โปรแกรม DOSKEY

การเก็บประวัติคำสั่งที่เคยเรียกใช้ไปแล้ว นับเป็นเรื่องที่ให้ประโยชน์ต่อผู้ใช้ได้มาก เพราะเราสามารถทราบว่าเคยใช้คำสั่งอะไรไปแล้วบ้าง ขณะเดียวกันสามารถเรียกคำสั่งเก่ามาแก้ไขและนำมาใช้ใหม่โดยไม่ต้องพิมพ์คำสั่งยาว ๆ อีก โปรแกรมจัดระบบงานยูนิกซ์และโปรแกรมจัดการฐานข้อมูลดีเบสก็มีลักษณะการใช้งานเก็บประวัติคำสั่งได้ ดังนั้นในดอส 5.0 จึงได้เพิ่มเติมโปรแกรม DOSKEY เข้ามาเพื่อเสริมการใช้งานด้านรับคำสั่งที่เครื่องหมายเตรียมพร้อมรับคำสั่งได้ดียิ่ง ๆ ขึ้น

โปรแกรม DOSKEY.COM ถือเป็นคำสั่งภายนอกที่อยู่ในรูปของไฟล์ เป็นโปรแกรมประเภทฝังในหน่วยความจำ ซึ่งใช้พื้นที่หน่วยความจำราว 4 กิโลไบต์ ในจำนวนนี้จะสงวนพื้นที่หน่วยความจำจำนวน 512 กิโลไบต์ ไว้สำหรับการเก็บคำสั่งและโปรแกรมมาโครต่าง ๆ โดยโปรแกรมมาโครจะประกอบด้วยคำสั่งดอสหนึ่งคำสั่งหรือคำสั่งดอสหลายคำสั่งมารวมกันในหน่วยความจำ เพื่อทำหน้าที่เหมือนแบตช์ไฟล์ ขนาดพื้นที่หน่วยความจำ 512 ไบต์นี้จะเก็บคำสั่งได้ราว 35 คำสั่ง ถ้าคำสั่งนั้นประกอบด้วยค่าเฉลี่ยของจำนวนตัวอักษรเท่ากับ 15 ตัวอักษร เราสามารถกำหนดขนาดบัฟเฟอร์หรือพื้นที่หน่วยความจำสำหรับเก็บคำสั่งได้เอง ค่าต่ำสุดจะเท่ากับ 256 ไบต์ และค่าดีฟอลต์เท่ากับ 512 ไบต์

เมื่อมีการเรียกใช้โปรแกรม DOSKEY นี้ครั้งแรก ดอสจะโหลดโปรแกรมนี้ฝังในหน่วยความจำ ทำให้สามารถเก็บประวัติการใช้คำสั่งต่าง ๆ และเก็บโปรแกรมมาโครที่สร้างขึ้นได้ ข้อดีของโปรแกรม DOSKEY รวมกับการใช้ปุ่มกดแก้ไขคำสั่ง ทำให้เราใช้งานคำสั่งเดิมได้ง่ายและสะดวกขึ้น นอกจากนี้การใช้ DOSKEY จะช่วยให้เราพิมพ์คำสั่งดอสหลาย ๆ คำสั่งเรียงติดต่อกันในบรรทัดคำสั่งเดียวได้อีกด้วย ถึงโปรแกรม DOSKEY จะให้ประโยชน์ได้มากแต่โปรแกรมนี้จะใช้พื้นที่หน่วยความจำส่วนหนึ่ง หากเราจำเป็นต้องใช้พื้นที่หน่วยความจำสำหรับงานอื่น ๆ ก็อาจต้องปลดโปรแกรม DOSKEY ออกเสียก่อนในบางกรณีและไปใช้ปุ่มกดแก้ไขคำสั่งเพียงอย่างเดียว

รูปแบบคำสั่ง DOSKEY

คำสั่ง DOSKEY จะมีรูปแบบคำสั่งดังนี้

```
DOSKEY [/REINSTALL] [/BUFSIZE=SIZE] [/MACROS] [/HISTORY]
       [/INSERT /OVERSTRIKE] [macroname=[text]]
```

โดยมีสวิตช์ให้เลือกใช้ดังต่อไปนี้

/REINSTALL สำหรับการโหลดเพื่อติดตั้งโปรแกรม DOSKEY ที่ฝังประจำในหน่วยความจำเพิ่มเติมอีกชุดหนึ่ง เราจะใช้สวิตช์ตัวเลือกนี้เมื่อต้องการใช้โปรแกรม DOSKEY กับการพร้อมรับคำสั่งของโปรแกรมไมโครซอฟต์วินโดว์

/BUFSIZE=size เราจะใช้เพื่อกำหนดขนาดหน่วยความจำสำหรับเก็บคำสั่งและโปรแกรมมาโครด้วยการระบุเป็นตัวเลขบอกจำนวนเป็นไบต์ค่าดีฟอลต์เท่ากับ 512 ไบต์ และค่าต่ำสุดเท่ากับ 256 ไบต์

/MACROS ใช้เพื่อการแสดงรายการโปรแกรมมาโครที่มีอยู่ในขณะนั้น เราอาจใช้สวิตช์นี้สั้นเป็น /M ก็ได้

/HISTORY ใช้เพื่อแสดงรายการคำสั่งเดิมที่เก็บไว้แล้วทั้งหมด อาจใช้สวิตช์นี้สั้นเป็น /H ก็ได้

/INSERT /OVERSTRIKE เมื่อจะใช้เพื่อกำหนดว่าข้อความที่นำเข้าใหม่จะไปแทนที่ข้อความเดิมหรือไม่ ถ้าเลือกใช้ /INSERT ข้อความใหม่จะไปแทนที่ในข้อความเดิมเหมือนกับการเลือกกดปุ่ม INS แต่ถ้าเราเลือกใช้สวิตช์ /OVERSTRIKE ข้อความใหม่จะทับข้อความเดิม ค่าดีฟอลต์ปกติจะเลือกใช้สวิตช์ /OVERSTRIKE

การเรียกโปรแกรม DOSKEY มาใช้งาน โดยไม่มีสวิตช์ตัวเลือกมาประกอบโปรแกรมนี้จะฝังอยู่ในหน่วยความจำ ใช้พื้นที่ 4128 ไบต์ (1020 ของเลขฐานสิบหก) ในจำนวนนี้จะมีขนาดบัฟเฟอร์เพื่อเก็บคำสั่งและโปรแกรมมาโคร 512 ไบต์ และอยู่ในโหมดพิมพ์ทับ /OVERSTIKE คำสั่งและจำนวนโปรแกรมมาโครได้มากน้อยเท่าไร จะขึ้นกับความสั้นยาวของคำสั่ง (ตามปกติในหนึ่งบรรทัดคำสั่งจะยอมให้เราป้อนตัวอักษรได้ไม่เกิน 127 ตัวอักษร) เราสามารถเพิ่มขยายบัฟเฟอร์สำหรับเก็บคำสั่งและโปรแกรมมาโครด้วยสวิตช์ /BUFSIZE=ขนาดพวกจำนวนเป็นไบต์ เช่น ถ้าต้องการขนาดบัฟเฟอร์เป็น 2 กิโลไบต์ จะเรียกใช้ด้วยคำสั่งดังนี้

```
DOSKEY /BUFSIZE=2048
```

เมื่อเรียกติดตั้งโปรแกรม DOSKEY แล้ว สามารถเรียกโปรแกรม DOSKEY มาติดตั้งซ้ำอีก ด้วยการใช้สวิตช์ /REINSTALL กรณีนี้จะใช้พื้นที่หน่วยความจำไปอีกราว 4 กิโลไบต์ และจะมีการใช้เก็บคำสั่งและโปรแกรมมาโครเป็นชุดใหม่

สวิตช์ /INSERT และ /OVERSTRIKE เป็นสวิตช์ที่เราต้องเลือกสวิตช์ใดสวิตช์หนึ่ง หากเราพอใจกับการใช้งานในโหมดแทรก ก็ต้องทำการติดตั้งเปลี่ยนจากโหมดพิมพ์ทับที่เป็นดีฟอลต์มาเป็นโหมดแทรกด้วยสวิตช์ /INSERT

โดยที่โปรแกรม DOSKEY นี้มีประโยชน์ต่อการใช้งานมาก อาจต้องเรียกติดตั้งทุกครั้งที่เริ่มเข้าเครื่อง ดังนั้นเราอาจบรรจุคำสั่ง DOSKEY ลงในไฟล์ AUTOEXEC.BAT ถ้าต้องการรันโปรแกรม DOSSHELL ด้วยคำสั่ง DOSKEY จะต้องนำมาก่อนโปรแกรม DOSSHELL

การเรียกประวัติคำสั่งมาใช้งาน

เมื่อทำการติดตั้งโปรแกรม DOSKEY ไว้ก่อน การพิมพ์คำสั่งดอสต่าง ๆ ที่เครื่องหมายพร้อมรับคำสั่งจะมีการเก็บบันทึกเอาไว้ เราสามารถเรียกคำสั่งต่าง ๆ เหล่านั้นมาแสดงและแก้ไข โปรแกรม DOSKEY ได้กำหนดปุ่มกดต่างดังรูปที่ 2 เป็นปุ่มเพื่อการเรียกคำสั่งที่เก็บไว้กลับคืนมา

รายการคำสั่งที่เคยเรียกใช้ไว้แล้ว ซึ่งมีการเก็บเอาไว้จะมีการจัดลำดับก่อนหลังไว้ด้วย เราสามารถเรียกรายการคำสั่งเหล่านั้นมาแสดง ด้วยการกดปุ่มฟังก์ชันคีย์ F7 โดยจะมีหมายเลขกำกับไว้ข้างหน้าดังรูปที่ 3 จากรูปที่ 3 นี้คำสั่งสุดท้ายที่เรียกใช้คือคำสั่ง time

ปุ่มกด	คำอธิบาย
ปุ่มลูกศรขึ้น	เพื่อเรียกคำสั่งก่อนหน้านี้จากในรายการคำสั่งมาแสดง
ปุ่มลูกศรลง	เพื่อเรียกคำสั่งถัดไปจากในรายการคำสั่งมาแสดง
F7	เพื่อเรียกรายการคำสั่งทั้งหมดมาแสดงพร้อมกันตามลำดับ โดยมีหมายเลขบอกลำดับด้วย
F8	ถ้าใช้ปุ่มที่โดด ๆ จะเป็นการแสดงคำสั่งก่อนหน้านี้จากในรายการคำสั่งมาแสดง แต่ถ้าใช้หลังการพิมพ์ตัวอักษรหรือชุดตัวอักษร จะเป็นการเรียกค้นคำสั่งที่เคยใช้ไปแล้ว ซึ่งเก็บอยู่ในรายการประวัติ เฉพาะที่ตรงกับตัวอักษรที่กำหนดไว้ก่อน เราสามารถกดปุ่มฟังก์ชันคีย์ F8 นี้อีกหลายครั้งให้มีการเรียกค้นวนถอยหลังต่อไปอีก
F9	เป็นการเรียกคำสั่งเดิมที่เคยใช้แล้ว ด้วยการระบุหมายเลขคำสั่งที่มีระบุไว้แล้วจากการใช้แสดงด้วยปุ่มฟังก์ชันคีย์ F7
ปุ่ม PgUp	เพื่อแสดงคำสั่งที่เก่าแก่ที่สุดในประวัติคำสั่ง
ปุ่ม PgDn	เพื่อแสดงคำสั่งล่าสุดในประวัติคำสั่ง
ESC	เพื่อการล้างหรือยกเลิกคำสั่งที่ปรากฏที่เครื่องหมายเตรียมพร้อมรับคำสั่ง
Alt-F1	การลบคำสั่งต่าง ๆ ที่เก็บเป็นประวัติไว้ออกทั้งหมด
Alt-F10	การล้างโปรแกรมมาโครที่เก็บไว้ออกทั้งหมด

รูปที่ 2 รายการปุ่มกดสำหรับเรียกประวัติคำสั่งต่าง ๆ ที่เก็บไว้มาแสดงหรือแก้ไข

```
C:\>
1: cls
2: dir/w
3: dir a:
4: chkdsk
5: chkdsk a:
6: date
7: time
C:\>
```

รูปที่ 3 การกดปุ่มฟังก์ชันคีย์ F7 เพื่อนำรายการคำสั่งที่เก็บไว้มาแสดง

จะเก็บอยู่ใน template เหมือนที่เราเคยใช้ในดอสเวอร์ชันเก่าด้วย ซึ่งจะเรียกคำสั่งล่าสุดนี้มาแสดงและแก้ไขด้วยปุ่มกดดังรูปที่ 1 คำสั่งที่เคยใช้ก่อนหน้าคำสั่งที่เจ็ด เราจะเรียกมาแสดงไม่ได้ด้วยปุ่มกดของเก่านั้น กรณีที่เราจะต้องใช้ปุ่มกดของ DOSKEY ดังรูปที่ 2 ที่กล่าวมาแล้ว

การกดปุ่มลูกศรขึ้นแต่ละครั้งจะช่วยให้เราเรียกคำสั่งเก่าย้อนหลังไปทีละคำสั่งได้ เนื่องจากมีจำนวนคำสั่งอยู่มาก และคำสั่งอาจไปหยุดที่คำสั่งใดในรายการก็ได้ ดังนั้นในระบบจึงต้องมีตัวชี้เพื่อบอกสถานภาพว่า ขณะนี้โปรแกรม DOSKEY ได้ชี้บอกที่คำสั่งหมายเลขเท่าใดอยู่ดังรูปที่ 4 เป็นตัวอย่างบอกตัวชี้แสดงด้วยเครื่องหมายมากกว่ากำลังชี้ที่คำสั่งหมายเลข 4 ถ้าเรากดปุ่มลูกศรขึ้นต่อไป ตัวชี้จะย้อนกลับไปที่คำสั่งหมายเลข 3 แต่ถ้าเรากดปุ่มลูกศรลง ตัวชี้จะเดินทางไปที่คำสั่งหมายเลข 5

กรณีที่มีจำนวนคำสั่งที่เก็บไว้เป็นจำนวนมาก ๆ การย้อนหรือเดินทางไปเรียกคำสั่งที่อยู่ห่างไกลออกไป อาจต้องกดปุ่มลูกศรขึ้นลงหลาย ๆ ครั้ง ซึ่งไม่ค่อยสะดวกนักเราสามารถเรียกรายการคำสั่งได้เร็วขึ้น ด้วยการใช้ปุ่มฟังก์ชันคีย์ F9 ที่จะให้เราระบุหมายเลขคำสั่ง สั่งกระโดดไปคำสั่งที่ต้องการทันที นอกจากนี้ปุ่มกด PgUp และ PgDn จะช่วยให้เราสามารถกระโดดไปเรียกใช้คำสั่งต้นสุดหรือเก่าแก่ที่สุด และคำสั่งล่าสุดหรือท้ายสุดได้ตามลำดับ

ปุ่มฟังก์ชันคีย์ F8 ถ้านำไปใช้โดด ๆ จะเหมือนปุ่มลูกศรขึ้น แต่ปุ่มกดนี้จะใช้เพื่อการค้นหามากกว่า เช่น ถ้าเรากดปุ่ม d ตามด้วยปุ่มฟังก์ชันคีย์ F8 เราจะได้คำสั่ง date แต่ถ้าเรากดปุ่ม i ต่อจาก d เป็น di ตามด้วยปุ่มฟังก์ชัน F8 เราจะได้คำสั่ง dir a: เป็นต้น

ปุ่มร่วมด้วยการกดปุ่ม Alt กับปุ่มฟังก์ชันคีย์ F7 จะเป็นการล้างคำสั่งทั้งหมด และปุ่ม ESC จะใช้เพื่อการยกเลิกคำสั่งที่ปรากฏที่ตำแหน่งพร้อมรับคำสั่ง

```
C:\>
1:  cls
2:  dir/w
3:  dir a:
4:  >chkdsk
5:  chkdsk a:
6:  date
7:  time
C:\>
```

รูปที่ 4 รายการคำสั่งที่เก็บไว้ในประวัติ แสดงตัวชี้คำสั่งกำลังอยู่ที่หมายเลขคำสั่งที่ 4

ปุ่มกด	คำอธิบาย
ปุ่มถอยกลับหรือ BS	ใช้ลบตัวอักษรที่อยู่ทางซ้ายของเคอร์เซอร์
Ctrl-End	ใช้ลบตัวอักษรทุกตัวไปจนจบบรรทัด โดยเริ่มจากตัวอักษรที่ตำแหน่งเคอร์เซอร์ขณะนั้น
Ctrl-Home	ใช้ลบตัวอักษรทุกตัวตั้งแต่เริ่มต้นบรรทัดไปจนถึงตัวอักษรที่ตำแหน่งเคอร์เซอร์ โดยไม่รวมตัวอักษรที่เคอร์เซอร์
Ctrl-ปุ่มลูกศรซ้าย	ใช้ย้ายเคอร์เซอร์ถอยกลับไปอักษรตัวแรกของคำที่อยู่ก่อนหน้า
Ctrl-ปุ่มลูกศรขวา	ใช้ย้ายเคอร์เซอร์เดินหน้าทีละคำ โดยจะไปปรากฏที่ตัวอักษรตัวแรกของคำถัดไป
Del	ใช้ลบตัวอักษรที่ตำแหน่งเคอร์เซอร์
End	ใช้ย้ายเคอร์เซอร์ไปตำแหน่งท้ายของคำสั่ง
Esc	ใช้ล้างหรือยกเลิกคำสั่งที่แสดงในขณะนั้น
Home	ใช้ย้ายเคอร์เซอร์ไปตำแหน่งต้นสุดของคำสั่ง
Ins	ใช้เปลี่ยนระหว่างโหมดการแทนที่และโหมดพิมพ์ทับ
ปุ่มลูกศรซ้าย	ใช้ย้ายเคอร์เซอร์ถอยกลับไปทางซ้ายหนึ่งตัวอักษร
ปุ่มลูกศรขวา	ใช้ย้ายเคอร์เซอร์เดินหน้าไปทางขวาหนึ่งตัวอักษร

รูปที่ 5 ปุ่มกดแก้ไขคำสั่งใน DOSKEY

การแก้ไขและเรียกใช้คำสั่ง

ในหัวข้อที่แล้ว เราได้กล่าวถึงการเรียกคำสั่งเดิมมาปรากฏที่ตำแหน่งรับคำสั่ง เราอาจต้องทำการแก้ไขคำสั่งเดิมก่อนนำไปเรียกใช้งานต่อ เช่น เราอาจเปลี่ยนชื่อไดเรคทอรีหรือชื่อไฟล์เสียใหม่ เป็นต้น โปรแกรม DOSKEY มีปุ่มกดสำหรับทำการแก้ไขเพิ่มเติมเข้ามาอีกหลายปุ่มดังรูปที่ 5 ซึ่งวิธีการใช้ปุ่มกดต่าง ๆ เหล่านี้จะคล้ายกับที่ภายในโปรแกรมเอดิเตอร์หรือโปรแกรมเวิร์ดโปรเซสเซอร์ทั่วไป จึงไม่ขออธิบายรายละเอียด

การสร้างแบตช์ไฟล์จากรายการคำสั่ง

รายการคำสั่งต่าง ๆ ที่เราเรียกใช้ถ้านำมาเรียงต่อ ๆ กันเก็บเข้าเป็นแบตช์ไฟล์นามสกุล .BAT ก็สามารถนำมาใช้งานได้ภายหลัง โดยไม่ต้องพิมพ์คำสั่งเข้าไปใหม่ จากคำสั่ง DOSKEY

พร้อมสวิตช์ /HISTORY จะเป็นการแสดงรายการคำสั่งที่เก็บไว้ทั้งหมด เราอาจเพิ่มการเปลี่ยนทิศทางแทนที่จะออกผลที่จอภาพ แต่จะให้ไปเก็บในไฟล์ด้วยการเติมเครื่องหมายมากกว่าและชื่อไฟล์ เนื่องจากในแบตช์ไฟล์นั้นเราอาจต้องเพิ่มเติมคำอธิบายหรือดัดแปลงคำสั่งบ้าง เราก็สามารถเรียกไฟล์นั้นมาแก้ไขใหม่ด้วยโปรแกรมเอดิเตอร์ต่าง ๆ ตัวอย่างของคำสั่งการสร้างแบตช์ไฟล์อาจใช้ดังนี้

```
DOSKEY /HISTORY >SAMPLE.BAT
```

โดยก่อนที่จะมีการเก็บคำสั่งใด ๆ ควรมีการสร้างคำสั่งเดิมออกหมดด้วยการใช้ปุ่ม Alt คู่กับฟังก์ชันคีย์ F7

การใช้คำสั่งมากกว่าหนึ่งคำสั่งในบรรทัดเดียว

ตามปกติเมื่อพิมพ์คำสั่งดอสเราจะใช้ได้เพียงคำสั่งเดียว แต่ถ้าเราติดตั้ง DOSKEY แล้ว เราสามารถพิมพ์คำสั่งดอสหรือเรียกโปรแกรมอื่น ๆ มากกว่าหนึ่งคำสั่งในบรรทัดคำสั่งนั้น โดยคำสั่งที่เรียงติดต่อกันได้นั้นจะต้องคั่นแบ่งคำสั่งด้วยการกดปุ่ม Ctrl พร้อมปุ่มตัวอักษร T (Ctrl-T) ซึ่งที่จอภาพจะแสดงด้วยตัวอักษรพิเศษ π คล้ายตัวอักษรพายทางวิทยาศาสตร์ ทุกครั้งที่เรากดปุ่ม Ctrl-T ตัวอย่างคำสั่ง

```
COPY C:\TEMP\TEST.TXT A: π DIR A: π
DEL C:\TEMP\TEST.TXT
```

จะเป็นคำสั่งคัดลอกไฟล์ไปยังไดรฟ์ A: แล้วตรวจสอบดูรายการไฟล์ในไดรฟ์ A: จากนั้นจึงทำการลบไฟล์ที่คัดลอกแล้วจากไดรฟ์ C:

เราสามารถพิมพ์คำสั่งบนบรรทัดคำสั่งในยาวถึง 128 ตัวอักษร ดังนั้นเราจะเรียงต่อคำสั่งต่าง ๆ ได้มากทีเดียว จนอาจเป็นเหมือนแบตช์ไฟล์ได้ทีเดียว

การสร้างโปรแกรมมาโคร

โปรแกรมมาโครเป็นชุดของคำสั่งที่เราสามารถเรียกมาทำงานด้วยการพิมพ์ชื่อที่กำหนดลักษณะเช่นเดียวกัน แบตช์ไฟล์ที่เราระบุเพียงชื่อของไฟล์ โปรแกรมมาโครจะมีวิธีการแตกต่างกับแบตช์ไฟล์ดังนี้

– โปรแกรมมาโครจะเก็บในหน่วยความจำ จึงทำงานได้รวดเร็วกว่าแบตช์ไฟล์ที่เก็บอยู่ในดิสก์

– เราสามารถเรียกมาโครมาทำงานได้ทันทีไม่ว่าเรากำลังใช้งานอยู่ไดเรคทอรีย่อยใดก็ตาม

– ในการเริ่มต้นบูตเครื่องใหม่ ส่วนมาโครที่อยู่ในหน่วยความจำชั่วคราวจะสูญหายหมด หากต้องการให้มาโครที่กำหนดไว้แล้วสามารถนำไปใช้ในภายหลัง ควรเก็บรายการคำสั่งสร้างมาโครไว้ในไฟล์ และพร้อมให้เรียกนำมาใช้ในภายหลังได้

– โปรแกรมมาโครที่สร้างด้วย DOSKEY จะต้องบรรจุคำสั่งทั้งหมดในบรรทัดคำสั่งบรรทัดเดียว ซึ่งอาจมีหลายคำสั่งที่เรียงต่อกัน คั่นระหว่างคำสั่งด้วยตัวอักษรพิเศษ ความยาวของตัวอักษรภายในบรรทัดจะต้องไม่มากกว่า 127 ตัวอักษร

– ขณะที่เราเรียกแบตช์ไฟล์มาทำงานและต้องการหยุดการทำงานนั้น เราจะกดปุ่ม Ctrl-C เพียงคราวเดียวเพื่อหลุดจากโปรแกรม แต่ถ้าเป็นโปรแกรมมาโครต้องกดปุ่ม Ctrl-C ทุก ๆ คำสั่งที่บรรจุในบรรทัดมาโครนั้น

– สามารถใช้คำสั่งกระโดด GOTO ในแบตช์ไฟล์เรียกแบตช์ไฟล์อื่นซ้อนภายในแบตช์ไฟล์ แต่ถ้าเป็นโปรแกรมมาโครแล้ว เราใช้คำสั่ง GOTO ไม่ได้ และเรียกมาโครซ้อนภายในมาโครไม่ได้ นอกจากนี้ในมาโครเราเรียกใช้คำสั่ง ECHO OFF เพื่อซ่อนรายการคำสั่งไม่ได้

– เราสามารถสร้างตัวแปรเป็นพารามิเตอร์สำหรับการแทนค่าทั้งในโปรแกรมมาโครและแบตช์ไฟล์พารามิเตอร์ของแบตช์ไฟล์เป็น %1 ถึง %9 แต่ในมาโครจะขึ้นต้นด้วย $1 หรือ $9

– ในโปรแกรมมาโครสามารถใช้หลักการเปลี่ยนทิศทางอุปกรณ์เหมือนเช่นในดอสหรือใช้ในแบตช์ไฟล์เพียงแต่ใช้สัญลักษณ์ต่างกัน คือ

$L หรือ $l เป็นการเปลี่ยนทิศทางอินพุต ใช้เหมือนเครื่องหมาย < ในดอส

$G หรือ Gg เป็นการเปลี่ยนทิศทางเอาต์พุต ใช้เหมือนเครื่องหมาย > ในดอส

GG หรือ gg เป็นการเปลี่ยนทิศทางเอาต์พุตแบบไปซ้อนต่อท้ายไฟล์ ใช้เหมือนเครื่องหมาย >> ในดอส

$B หรือ $b เป็นการเปลี่ยนจากเอาต์พุตหนึ่งไปเป็นอินพุตหนึ่งที่เรียกว่าท่อใช้เหมือนเครื่องหมาย Vertical bar ในดอส

– เราสามารถเรียกโปรแกรมแบตช์ไฟล์จากภายในมาโคร แต่ไม่สามารถเรียกมาโครจากภายในแบตช์ไฟล์ เราจะทำการได้บรรรจุคำสั่งสร้างมาโครภายในแบตช์ไฟล์

– ภายในมาโคร เราสามารถกำหนดตัวแปรสภาวะแวดล้อม (enviroment variable) ขึ้นมาได้ แต่เราจะนำมาใช้

อ้างกับคำสั่งต่าง ๆ ไม่ได้ เช่น การกำหนดตัวแปรหนึ่งเป็นช่องไดรฟ์และไดเรคทอรีหนึ่งด้วยคำสั่ง Set name=C:\TEMP เราจะเรียก ใช้คำสั่ง dir name ในมาโครไม่ได้ แต่ใช้ในแบตช์-ไฟล์ได้

การสร้างมาโครจะเป็นการกำหนดด้วยการใช้คำสั่ง DOS-KEY แล้วตามด้วยชื่อมาโคร ตามด้วยเครื่องหมายเท่ากับและคำสั่งที่เราต้องการเขียนย่อ ตัวอย่างเช่น

```
DOSKEY CTOA=COPY C:\TEMP\*.TXT A:
```

เป็นการกำหนดมาโครชื่อ CTOA ที่ประกอบด้วยคำสั่งเดียวของการคัดลอกไฟล์จากช่องไดรฟ์ C ไปยังไดรฟ์ A เวลาเรียกใช้มาโครนี้จะเป็นการพิมพ์ชื่อ CTOA ที่เครื่องหมายพร้อมรับคำสั่ง

กรณีถ้ามาโครประกอบด้วยคำสั่งมากกว่าหนึ่งคำสั่ง คำสั่งเหล่านั้นจะแบ่งตอนแยกจากกันด้วยตัวอักษร $t ตัวอย่างเช่น

```
DOSKEY mv=COPY *.TXT C:\TEXT $T dir C:\TEXT
```

คำสั่งข้างต้นเป็นการกำหนดมาโครชื่อ mv ซึ่งจะทำการคัดลอกไฟล์นามสกุล .TXT จากไดเรคทอรีปัจจุบันไปยังไดเรคทอรี \TEXT ของไดรฟ์ C จากนั้นแทรกกลางด้วยปุ่ม $T แล้วจึงทำงานตามคำสั่งที่สองเพื่อแสดงรายการไฟล์ในไดเรคทอรี \TEXT

ถ้าหากมีการกำหนดชื่อมาโครเหมือนคำสั่งดอส ระบบดอสจะเรียกชื่อมาโครมาทำงานก่อน หากเราต้องการเรียกคำสั่งดอสนั้นจริง ๆ เราอาจแทรกช่องว่างหน้าคำสั่ง โดยให้เว้นช่องว่างก่อน แล้วจึงตามด้วยชื่อคำสั่งนั้น ดังเช่นตัวอย่างนี้

```
DOSKEY DIR=dir/w
```

กรณีนี้เรามีทั้งมาโครชื่อ DIR และคำสั่งดอส ถ้าเรียกด้วยคำสั่ง DIR ตามด้วย Enter เราจะได้รายการไฟล์อย่างย่อ ๆ ตาม dir/w แต่ถ้าพิมพ์ช่องว่าตามด้วย dir เราก็จะได้รายการไฟล์รูปแบบเต็มตามคำสั่งดอส DIR

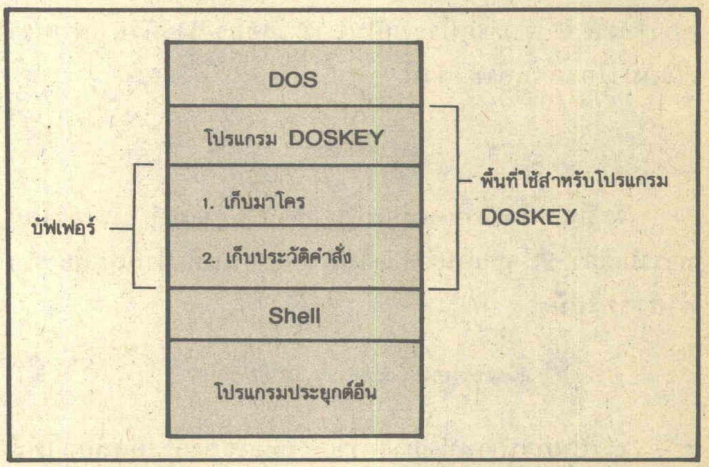
การแก้ไข เซฟและลบมาโคร

โปรแกรมมาโครที่เราสร้างไว้แล้ว สามารถเรียกนำมาดูได้ด้วยการเรียกคำสั่ง DOSKEY ตามด้วยสวิตช์ /MACROS หรือ /M หากต้องการแก้ไขเปลี่ยนแปลงก็สามารถเรียกคำสั่งที่เราสร้างขึ้นมาแก้ไข ด้วยการใช้ปุ่มกดแก้ไขประวัติคำสั่งตามที่กล่าวมาแล้วทั้งหมด ส่วนการลบมาโครที่ไม่ต้องการออกเราจะใช้คำสั่ง DOSKEY ตามด้วยชื่อมาโครนั้น แล้วตามด้วยเครื่องหมายเท่ากับ แต่ไม่พิมพ์ข้อความของคำสั่งใด ๆ ต่อท้าย โปรแกรม DOSKEY จะลบมาโครนั้นออกจากหน่วยความจำ ดังตัวอย่างการลบมาโคร DOSKEY dir= ถ้าต้องการลบมาโครทั้งหมดออกจากหน่วยความจำ เราจะใช้ปุ่ม Alt คู่กับปุ่มฟังก์ชันคีย์ F10 ซึ่งการลบมาโครนี้จะลบเฉพาะที่หน่วยความจำเก็บมาโคร แต่ไม่มีผลไปถึงการลบรายการคำสั่งที่สร้างมาโครนั้น

วิธีการจัดการหน่วยความจำของโปรแกรม DOSKEY เพื่อเก็บคำสั่งและมาโครอาจแสดงได้ดังรูปที่ 6 โดยในช่วงแรกจะโหมดโปรแกรม DOSKEY และในส่วนที่เป็นบัฟเฟอร์ขนาด 512 ไบต์ (ตามค่าดีฟอลต์ไม่มีการเพิ่มหรือลดขนาด) จะใช้เก็บประวัติเก็บคำสั่งต่าง ๆ ถ้ามีการสร้างโปรแกรมมาโครขึ้นมา บัฟเฟอร์ส่วนแรกจะนำมาใช้เก็บโปรแกรมมาโครและใช้เก็บคำสั่งต่าง ๆ ที่ใช้ต่อจากการกำหนดมาโคร คำสั่งเดิมที่เคยใช้ก่อนที่จะมีการกำหนดมาโครจะย้ายไปอยู่ตอนท้ายของบัฟเฟอร์ ทำให้บัฟเฟอร์แบ่งเป็นสองตอนดังรูปที่ 6 ถ้ามีการใช้สวิตช์ /BUFSIZE เพิ่มขนาดบัฟเฟอร์ ส่วนโปรแกรมจะยังคงเดิมและส่วนบัฟเฟอร์จะเพิ่มขึ้นตามที่ได้กำหนดขึ้น

โปรแกรมมาโครที่สร้างขึ้นอยู่เฉพาะในหน่วยความจำ ถ้าปิดเครื่องโปรแกรมมาโครต่าง ๆ ก็จะถูกล้างหมดไปเอง มาโครต่าง ๆ เหล่านี้อาจมีประโยชน์เรียกใช้บ่อย ๆ ในภายหลัง กรณีนี้เราควรเก็บมาโครไว้ในแบตช์ไฟล์ แล้วเรียกแบตช์ไฟล์มาใช้ภายหลังการเก็บโปรแกรมมาโครในแบตช์ไฟล์ จะเรียกด้วยคำสั่ง

```
DOSKEY /MACROS >MACRO.BAT
```



รูปที่ 6 หน่วยความจำแสดงการจัดการหน่วยความจำของโปรแกรม DOSKEY

แต่เนื่องจากคำสั่ง DOSKEY /MACROS จะแสดงเฉพาะชื่อมาโครและคำสั่งในมาโคร ดังนั้นเราต้องนำไฟล์ MACRO.BAT มาเพิ่มเติมคำสั่ง DOSKEY นำหน้าทุกรายการเสียก่อนดังรูปที่ 7 ทุกครั้งที่เรารันแบตช์ไฟล์ MACRO.BAT ก็จะมีการโหลดโปรแกรมมาโครนั้นลงสู่หน่วยความจำ

```
รายการมาโครที่ได้จากคำสั่ง DOSKEY /MACROS >MACRO.BAT
CTOA=COPY C:\TEMP\*.TXT A:
mv   =COPY *.TEXT C:\TXT $T dir C:\TEXT
DIR  =dir/w
นำแบตช์ไฟล์ MACRO.BAT มาแก้ไขเพิ่มเติมข้อความ DOSKEY แล้วเซฟไฟล์ใหม่ได้ผลดังนี้
doskey CTOA=COPY C:\TEMP\*.TXT A:
doskey mv  =COPY *.TXT C:\TEXT $T dir C:\TEXT
doskey DIR =dir/w
```

รูปที่ 7 การเซฟรายการมาโครใส่แบตช์ไฟล์

การสร้างมาโครโดยมีพารามิเตอร์แทนที่

เราสามารถสร้างโปรแกรมมาโคร โดยใช้พารามิเตอร์แทนที่เหมือนการใช้พารามิเตอร์แทนที่ในแบตช์ไฟล์ โดยในแบตช์ไฟล์นั้นจะใช้สัญลักษณ์ %1 จนถึง %9 แต่ในมาโครจะใช้สัญลักษณ์ $1 จนถึง $4 ดังตัวอย่างเช่น

```
doskey findfile=dir C:\ $1
```

เป็นการสร้างมาโครชื่อ findfile ซึ่งจะเป็นการขอดูรายการไฟล์ที่ปรากฏในไดเรคทอรีรากหรือไดเรคทอรีย่อยอื่นของไดรฟ์ C ตามชื่อที่ระบุเป็นพารามิเตอร์ $1 ดังนั้นหากเราเรียกมาโครด้วยคำสั่งดังนี้

```
findfile test.*
```

จะมีความหมายเหมือนกับ dir C:\test.* นั่นเองโดยพารามิเตอร์ $1 จะแทนที่ด้วยไฟล์ test.* ทันที ทำนองเดียวกับคำสั่งต่อไปนี้

```
doskey mv=copy $1 $2 $t del $1
```

จะเป็นการกำหนดมาโครชื่อ mv ของการคัดลอกไฟล์ต้นทาง $1 ไปเป็นไฟล์ปลายทาง $2 แล้วต่อด้วยการลบไฟล์ต้นทาง $1 การสั่งด้วยคำสั่ง

```
mv test1.txt test2.txt
```

จะหมายถึง การคัดลอกไฟล์ COPY test1.txt test2.txt ตามด้วยการลบไฟล์ del test1.txt นั่นเอง

นอกจากนี้เราสามารถสร้างโปรแกรมมาโครที่ใช้พารามิเตอร์แทนที่ $* มากำหนดข้อความต่อท้ายคำสั่งมาโครเป็นพารามิเตอร์เดียว ซึ่งตามปกติแล้วพารามิเตอร์ของดอสจะคั่นด้วยช่องว่าง การใช้ $* หลังคำสั่งมาโครจะทำให้ข้อความที่มีช่องว่างเป็นพารามิเตอร์เดียว ตัวอย่างเช่น การกำหนดมาโคร d เป็น

```
doskey d=dir $*
```

เราสามารถเรียกมาโครได้หลายรูปแบบเช่น

```
d *.txt          $* จะแทนด้วย *.txt
d *.txt /s       $* จะแทนด้วย *.txt /s
d *.txt /s /b .$* จะแทนด้วย *.txt /s /b
```

สรุป

โปรแกรม DOSKEY ที่เพิ่มเติมเข้าในดอส 5.0 ได้เสริมขีดความสามารถให้เราเรียกใช้คำสั่งเดิมที่เก็บเป็นประวัติได้ดีเด่นมากขึ้น เรามีปุ่มกดเรียกคำสั่งมาดัดแปลงแก้ไขได้มากขึ้นโดยไม่ต้องเสียเวลาพิมพ์คำสั่งเข้าไปใหม่ทั้งหมด เราสามารถพิมพ์คำสั่งให้ซ้อนเรียงติดต่อกันในบรรทัดเดียวได้ยาวถึง 127 ตัวอักษร และยังสามารถสร้างเป็นโปรแกรมมาโครในลักษณะของแบตช์ไฟล์ได้อีกด้วย รูปแบบของโปรแกรมมาโครนี้จะทำงานได้เร็วกว่าแบตช์ไฟล์ เพราะคำสั่งต่าง ๆ นั้นได้เก็บอยู่ในหน่วยความจำแล้ว ทำให้เรียกใช้ได้ง่ายจากในทุก ๆ ไดเรคทอรี ไม่จำเพาะต้องอยู่ในไดเรคทอรีที่เก็บไฟล์เหมือนแบตช์ไฟล์

โปรแกรม DOSKEY เป็นอีกส่วนหนึ่งที่เราผู้ใช้จะได้ประโยชน์มากขึ้น อยากให้ผู้อ่านหาดอส 5.0 มาใช้งาน เพราะมีจุดเด่นมากกว่าดอสเวอร์ชันเก่าหลายอย่างทีเดียว